AMPS Win32 Solutions Guide

Copyright © 2021 by Moody's Analytics, Inc.

All Rights Reserved.

Moody's Analytics provides this documentation without warranty of any kind, either express or implied, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose.

AMPS, AMPS Lan, AMPS Net, AMX, COGENT, ENQUE and Interceiver are trademarks of Moody's Analytics, Inc.

Other brands and product names used in this manual may be registered trademarks or trademarks of their respective holders.

June 2021

Table of Contents

Introduction

Overview of the Solutions Manual

This is the Win32 Server Solutions Guide for AMPS. This guide is designed to answer the question "I've successfully installed AMPS. What are my next steps?" It describes different ways that the AMPS software can be used and integrated with your system. You may choose to use one, several or none of the solutions in this manual.

Note: This document discusses various ways to make use of the news and information provided by AMPS. Clearly, these are not the only solutions possible.

Organization of This Guide

You will need a basic understanding of the AMPS system. This is provided in the *Overview of the AMPS for Win32* section. Also, refer to documents in the *Related Documentation* section for further information on *AMPS*.

The *Overview of Win32 Server Solutions* section provides a brief overview of the various solutions discussed within this guide. You should read through this section and choose the solutions you would like to understand in more detail. You can then go to the major section on the chosen solutions and read more detailed information about how to implement the solution at your location.

Related Documentation

- AMPS Win32 Diagnostic User Manual. *This document describes the software used to diagnose an AMPS connection.*

Overview of *AMPS for Win32*

The AMPS software consists of:

- AMPS Interceiver: This is a set of Microsoft Windows DLLs that maintains a connection between your system and the AMPS server. This connection is used to transmit news and other information to your system.
- AMPS XML Service: This service uses the Interceiver to get news and information from the AMPS server. It then provides a generic interface whereby one or more external programs (applications or Win32 services) can receive and process the news.
- One or more applications that process the XMLNews stories and associated metadata files (RDF files).

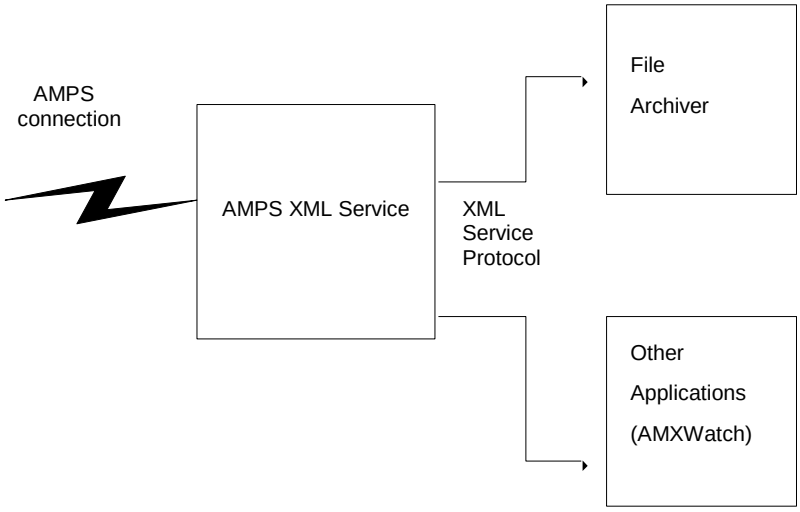


Figure 1: Overview of AMX Components

As shown in Figure 1, the AMPS XML Service receives news and information from the AMPS server. One or more applications may then receive the information from the AMPS XML Service using the XML Service protocol. These applications may be Win32 Services or standard Win32 applications.

You may write these applications, or you may choose to use applications that are provided by NewsEdge Customer Support. The File Archiver is an application that connects to the AMPS XML Service. This service is described in more detail in this document. The news is transmitted to the attached applications formatted in implementations of the NITF XML News DTD and the metadata RDF DTD formats.

Overview of Win32 Server Solutions

AMPS File Archiver Service

The AMPS File Archiver provides the ability to save news into a directory. It can also convert the XMLNews stories to a user-defined HTML version at the same time. The resulting set of directories containing news stories would then be used as input to another program. For example, you may want to index them using a full-text search engine such as Microsoft Index Server. You might also create a process that periodically sweeps the directory and moves articles to another directory where they may be processed by another program specific to your needs.

Use the AMPS File Archiver Service if you would like to have the news written to a single directory or a hierarchy of directories. Optionally, you may also have the XML stories translated to some format other than XML (e.g., HTML). You would then write additional software to provide the appropriate access to the news stories for your application.

AMPS AMXWatch Application

The AMPS utility AMXWatch is a tool that provides the ability to examine the incoming news as it arrives at the server. This application keeps a list of the last 200 news story files that have arrived and can display the 'raw' text of these files.

AMX Service Client DLL API

The AMX Service Client DLL API provides a library of routines to simplify the task of building custom applications that integrate with the AMX Service. If you are writing applications in C or C++, you can use the AMX Service Client DLL API to simplify the task of connecting to the AMX Service and receiving the XML news.

Use the AMX Service Client DLL API if you want to write a custom C or C++ application to process XMLNews stories directly. You will need to be an experienced C or C++ developer to use this solution.

Summary of Solutions

The following matrix summarizes information for the above solutions:

Program	General Description	Degree of Solution	Also requires:	Installation file
AMX File Archiver	Write XML, RDF and/or HTML files to disk	Requires front-end development to display and/or index news		AMXFileArchiver.exe

AMX Service Client DLL API	Provides programming API to connect to the XML Service and collect stories	Requires additional C development to use.		AMXCToolkit.exe

AMX Solutions – Solution Format

Each solution is described with the following headings:

1. General Description
2. Installation
3. Use Instructions

Firewall/DMZ

General Introduction

In general, IT departments do not like having software that connects out to the Internet and brings data back to the 'inside' of their firewall. There are several ways to use the File Archiver and XML Service to get AMPS data inside a firewall and still maintain security. We outline one solution. There may be others.

Installation

You need two machines for this solution. The first machine, the server that the File Archiver and XML Service run on, is the AMX server. This machine is located 'outside' in a DMZ. The AMX server needs to be able to pass the outside firewall and make TCP/IP connections on port 80 and 443, which access the AMPS server.

The second machine, the REPOSITORY, can be 'on the inside' but it needs to be reachable from the AMX server. The REPOSITORY server makes a 'shared folder,' controlled through a domain account, visible to the network. A domain account is needed to use the share. This account needs to have just the rights to write to the 'shared folder.'

Install the XML Service and File Archiver on the AMX server in the normal way. After the installation completes, use the Services Applet (Control Panel->Administrative Tools ->Services) to stop the File Archiver service. Open the properties of the File Archiver service and use the LogOn tab to change the account used for running the File Archiver service. Change the File Archiver to use the domain account that has rights to write to the 'shared folder' provided by the REPOSITORY server. Next, use the File Archiver configuration utility to change the root archive to \\REPOSITORY\shared_folder (the UNC name that points to the shared folder.) After making the changes, restart the File Archiver.

Use Instructions

The File Archiver will write the incoming stories to the REPOSITORY file share. Your application can use, move, or remove the incoming data as it requires.

Microsoft Index Server

General Introduction

This section describes one way to use the Microsoft Index Server to incorporate the HTML files created by the AMX File Archiver into your website. The sample query page for the Index Server was modified to test the searches described below.

Installation

1. You may need to change the configuration of the AMX Service using XMLSvcCfg. The value that may need to be changed is the "CompanyCode delimiter" value. This value must be a " " (space). This allows the Microsoft Indexer to recognize a company ticker symbol when searching on CompanyCode.
2. Configure the File Archiver to use Output directory type "Publication Date/time hierarchy" and Output File Type of "Template only". The “Output File Character Conversion” should be set to “HTML character entities”. This will create the "html" tree on the hard drive that will be used to create a virtual directory.
3. Make sure the template is adding metadata tags that will allow searches based on this data.
4. Using the Microsoft Management Console for Internet Information Server (IIS), create a virtual directory under the "Default Web Site" that uses the "{File Archiver Root output directory}\web\data\html" as its local path. The name of the virtual directory can be any allowable name under IIS. For this example, we will use the name "RealTimeNews".
5. At this point, if the Index Server is already running and indexing the default website, then the News Stories that have been saved to this point, as well as new stories, will be added to the Indexer Server Catalog. You can view the number of documents being added to the catalog by using the Index Server Manager and selecting "Index Server on Local Machine" in the left pane.
6. If the default catalog has been deleted, create a new one. Using the Index Server Manager, right-click on "Index Server on Local Machine" in the left pane. Then select the "New"/"Catalog" item. A dialog appears allowing you to configure the name of the catalog (any allowable name can be entered, such as "RealTimeNews") as well as the hard drive location of the catalog.
7. To allow searching for information provider/service and ticker symbols, the changes needed in the query.asp file found in the Index Server sample are as follows. Search for "CompSearch" in the file and find the line "Q.Query = CompSearch". Add the following lines:
    - Q.Catalog = "RealTimeNews" (if the catalog name is "RealTimeNews". If the catalog is the default, do not enter this line.)
    - Q.DefineColumn "companycode (DBTYPE_WSTR) = d1b5d3f0-c0b3-11cf-9a92-00a0c908dbf1 companyCode"

- Q.DefineColumn "providercode (DBTYPE_WSTR) = d1b5d3f0-c0b3-11cf-9a92-00a0c908dbf1 providerCode"
- Q.DefineColumn "servicecode (DBTYPE_WSTR) = d1b5d3f0-c0b3-11cf-9a92-00a0c908dbf1 serviceCode"
- Q.Columns = "DocTitle, vpath, filename, size, write, characterization, rank, companycode, providercode, servicecode"

Use Instructions

The following searches can now be done from the sample page.

- @providercode "num=14" ----- This search will return all PRNewswire stories. Note: the quotation marks are necessary to make the query legal.
- @companycode ibm ----- This search will return all stories that have the 'IBM' ticker symbol as part of the metadata.
- @DocTitle "AP Top " ----- This search will find all Associated Press "Top" stories.
- (@write > 2019/08/18) AND (@DocTitle "AP Top ") ----- This search finds all Associated Press "Top" stories with a date after August 18, 2019.

To add other metadata to the search, the template file must be changed to insert the metadata as <META> tags into the head portion of the HTML document. Then the Q.DefineColumn must be defined for the metadata and the name must be added to the Q.Columns list.

The template must have all RDF fields that could be used for searching in the HEAD portion of the HTML document as META tags like:

```
   <HEAD>
         <META http-equiv="content-type" content="text/html; charset=utf-
8">
         <?XNews foreach xn:companyCode?>
         <META name="companyCode" content="<?XNews insert
xn:companyCode?>" >
         <?XNews end?>
         <META name="providerCode" content="num=<?XNews insert
xn:providerCode?>" >
         <META name="serviceCode" content="num=<?XNews insert
xn:serviceCode?>" >
         <TITLE>
               <?XNews if xn:title?>
                     <?XNews insert xn:title ?>
               <?XNews else?>
                     NewsPack Article.
```

```
            <?XNews end?>
      </TITLE>
</HEAD>
```

This would allow CompanyCode, ProviderCode, and ServiceCode. Notice the "num=" for providerCode and ServiceCode. This is required because Microsoft Index Server ignores single character words like '1'.

For a tag that might not be present, such as SubjectCode, the following code is needed in the template file.

```
<?XNews if xn:subjectCode?>
      <META name="subjectCode" content="<?XNews insert
                  xn:subjectCode?>" >
<?XNews end?>
```

The search strings for these fields would be as follows:

@providerCode "num=14" would return all stories delivered by PRNewswire.

@companyCode ibm would return all stories that have 'IBM' as a CompanyCode.

There are some default "properties" for HTML files. Two important ones are "write" and "DocTitle".

(@Doctitle "AP Top") AND (@write > 2019/08/23) would return all stories written to disk after August 23, 2019 that have "AP Top" in the headline.

Now it is just a matter of some ASP code (query.asp is a great start) to FULLY integrate AMX directly into your website.

File Archiver Service

General Description

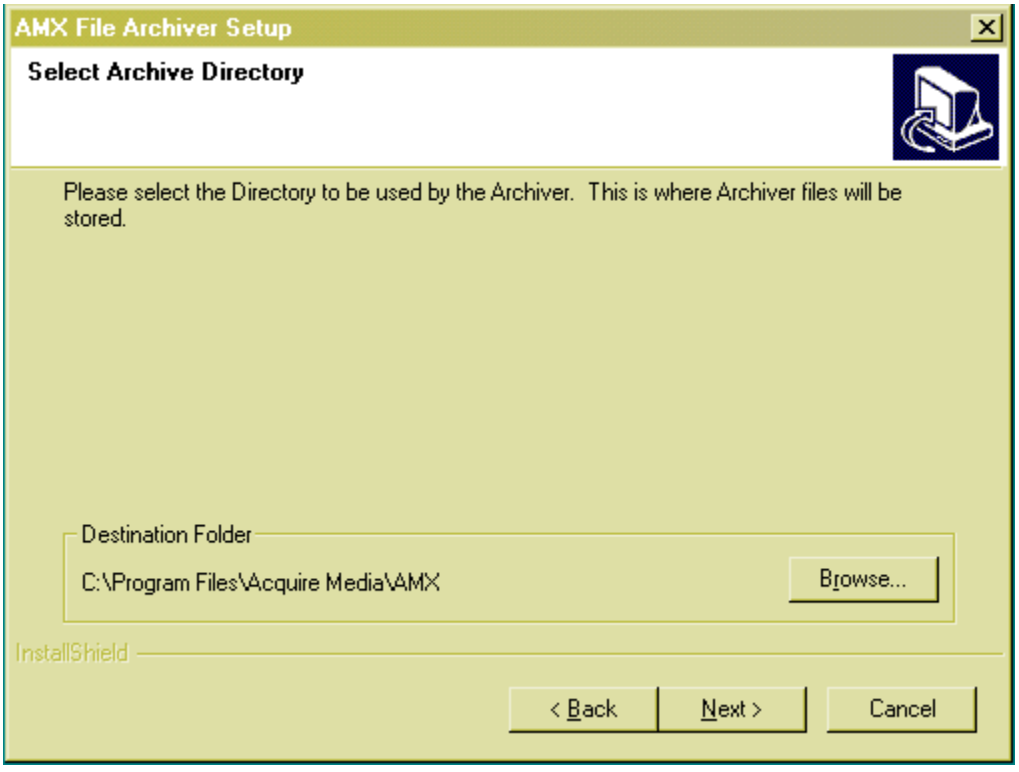
This section describes the File Archiver operation and configuration. The File Archiver is used to write XML and/or HTML files into a flat file system or a directory hierarchy. It is also used by the AMX Indexer as well as the Verity Spider and Microsoft Index Server solution described above.

Installation

The File Archiver is installed using the provided installation script. In special installation situations, the following may be applicable.

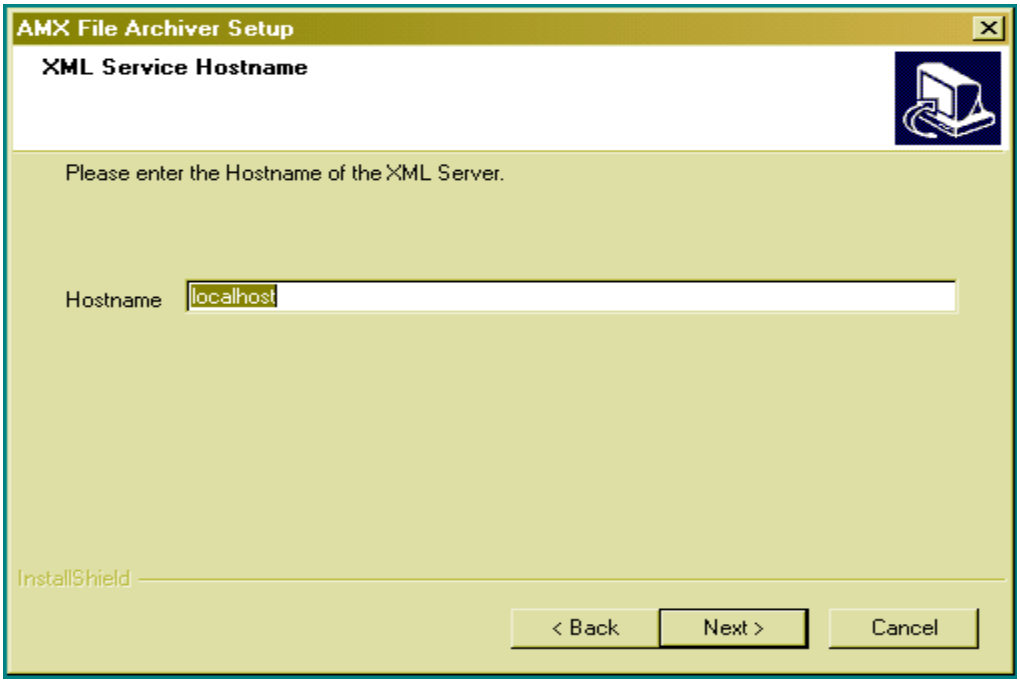
Initial Installation

During the installation you will see a dialog like the following:



You will need to select the directory where you are going to have the File Archiver write the files being received from the AMPS XML Service. This may be on a different drive than where the programs are stored. Be sure to choose a drive with adequate room (typically, several gigabytes). We **do not** recommend using drive 'C:'.

The next dialog you will see is:

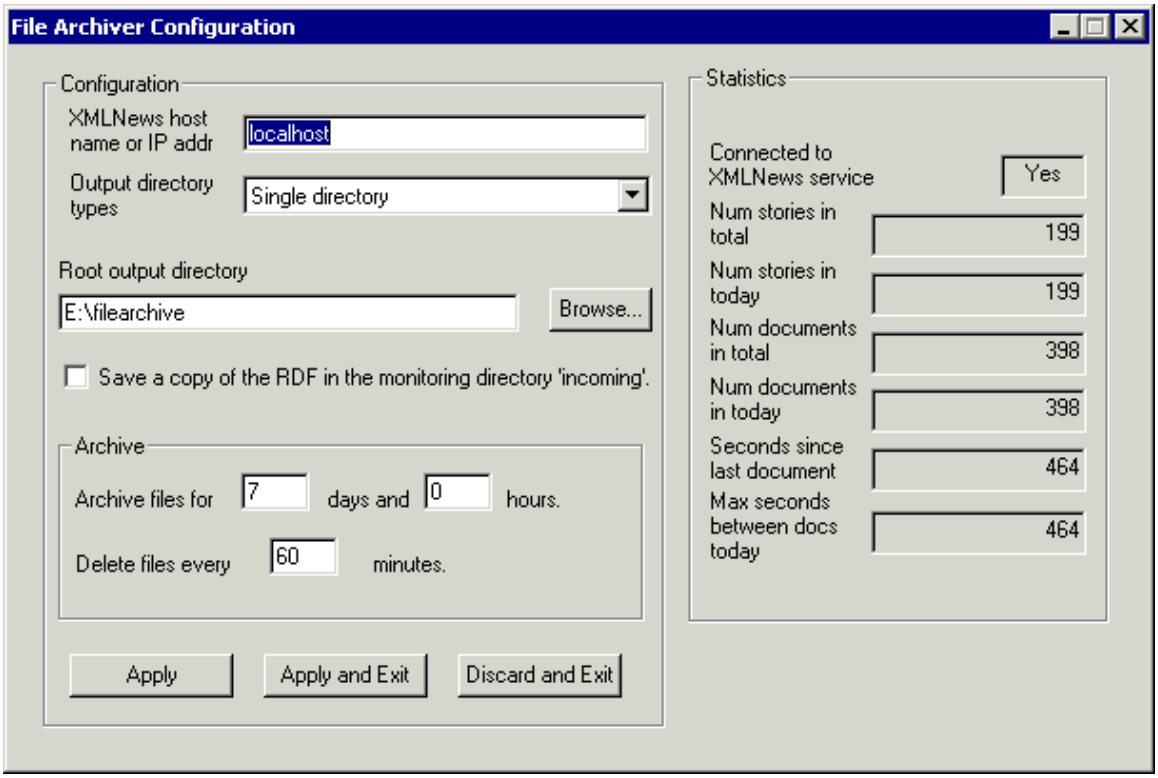


This dialog is used to specify the location of the host machine that is running the XML Service. Normally, if you are running the AMPS XML Service on the same machine as the File Archiver, you should accept the default “localhost”. Otherwise, you will need to type in the host name or the IP address of the host where the XML Service is running.

The other items in the installation are standard and should be familiar to individuals who have done Microsoft Windows installations in the past.

File Archiver Configuration Tool Usage

The following is the File Archiver configuration and statistics dialog. This dialog results after starting the ArchConfig.exe program ("File Archiver Status & Configuration" under "Acquire Media" in the program list). It communicates with the BaseArch.exe program.



Fields in the Configuration area:

XML News Host Name or IP Addr - Enter the "DNS name" or IP address of the machine that has the AMPS XML Service running on it. Examples are "localhost" or "127.0.0.1".

Root Output Directory - This selection defines the root directory where the files will be written. For example, the entry "c:\XMLNewsarch" combined with "Publication/Received Date/time hierarchy" and "Template only" would result in "c:\XMLNewsarch\web\data\html\{YYYYMMDD}\{HH}. There is a "Browse" button for convenience.

Fields in the Archive area:

Archive Files for "DD" days and "HH" hours. - The two fields define the total amount of time a story will remain on the disk before it is deleted. The valid range for days is 0-7 and the valid range for hours is 0-23. Entry of zero for both fields results in an error box indicating that there must be some archive time.

Delete Files every "MM" minutes. - The entry defines the time between cleanup operations of old files. The valid range for minutes is 15-120.

‘Apply’, 'Apply and Exit' and 'Discard and Exit' buttons - For the 'Apply' and 'Apply and Exit' buttons, all changes will take affect after selection of either button with one exception. Changing the AMX host currently requires the service to be restarted.

The Statistics area contains information you can use to monitor the service. Fields in the Statistics area:

Connected to XMLNews service - Yes or No. Defines whether the archiving process is connected or not connected to the AMX Service.

Num Stories in total - This is the count of stories processed by the archiving process since the start of the program.

Num Stories in today - This is the count of stories processed by the archiving process since Midnight GMT.

Num Documents in total - This is the count of documents processed by the archiving process since the start of the program. Documents include mostly XML and RDF files. The count will also include associated files.

Num Documents in today - This is the count of documents processed by the archiving process since Midnight GMT. Documents include mostly XML and RDF files. The count will also include associated files.

Seconds since last document - This is the number of seconds since the last document.

Max seconds between docs today - This is the maximum number of seconds between documents since Midnight GMT.

Appendix A – Story Identifier

Each XML story is tagged with a story identifier. This story identifier is used as the filename. The XMLNews story filename is the story identifier followed by the extension ".xml". The XMLMeta file name is the same story identifier followed by the extension ".rdf".

The format of the story identifier is as follows:

yyyymmddhhmmpppppppppssssssssuuuuuuuuuuuuu

This format is described in detail below.

- A date/time based on the publication date/time. The format for this is:

  yyyymmddhhmm

  Where yyyy is the four-digit year, mm is the two-digit month (January = 01, February = 02, and so on), dd is the two-digit day (e.g., 01, 02, …, 10, 11, …), hh is the hours (represented in 24-hour military time), and mm is the minutes. The time is in Eastern Time and reflects the publication time of the story.

- pppppppp is the 8-character information provider string. These strings map directly to specific providers. So, a provider will always have the same provider string.
- ssssssss is the 8-character information provider service string. These strings map directly to information provider designated services within the information provider's feed.
- uuuuuuuuuuuuu is a unique string added to the story identifier to ensure uniqueness of the identifier among stories sent by the same information provider in the same service at the same time. In some cases, the information provider provides the unique string. In other cases, the identifier is generated automatically by the system. Often, though not always, this will be a 13-character string.

To summarize:

- characters 1 through 12 are always the date
- characters 13 through 20 are always the provider designation
- characters 21 through 28 are always the service designation
- the remaining characters are a unique string to ensure uniqueness of the identifier